Título: Una vulnerabilidad en Google AdSense. Extracción automática de enlaces de anuncios.

Prof. Dr. Manuel Blázquez Ochando. Departamento de Biblioteconomía y Documentación. Facultad de Ciencias de la Documentación. Universidad Complutense de Madrid. manublaz@ucm.es

Resumen: Partiendo de técnicas de XSS (Cross Site Scripting) y de Web Crawler es posible superar las barreras del sistema de anuncios de Google Adsense, obteniendo los enlaces validados de los anuncios publicados en un sitio web. Este método implica la obtención del código fuente construido por el applet de java de Google para la publicación de anuncios, su manipulación y final recuperación de enlaces. Una vez obtenidos los enlaces de los anuncios, pueden aprovecharse las sesiones de los usuarios que visitan otras páginas web para cargar, en segundo plano, mediante re-direccionamiento sencillo, al marco de un iframe oculto dichos enlaces, de forma que las direcciones IP que efectúan los clics sean diferentes en cada caso.

Palabras clave: Google Adsense, Vulnerabilidad, Cross Site Scripting, XSS, anuncios, PPC, Pay per click

Introducción

La correcta encriptación y codificación de los anuncios publicados por terceros resulta un problema relevante para lograr la confianza y fiabilidad de los anunciantes en los sistemas de pago por clic. Con este método, la empresa anunciadora retribuye a las páginas web que publican sus anuncios, siempre y cuando los usuarios hagan clic en ellos, de forma proporcional y transparente a su número de visitas y validación. Este sistema es utilizado por Google Adsense, el servicio de publicidad de Google, que se encarga de gestionar los anuncios adecuados para cada página web y perfil de usuario. Al estudiar el sistema de anuncios de Google, cabe formular diversas preguntas ¿Cuál es el nivel de seguridad de los anuncios? ¿Pueden obtenerse los enlaces automáticamente? ¿Es posible hacer clic de forma automática en los anuncios obtenidos? En el artículo de (Mann, C.C. 2006) publicado en la revista Wired, se destaca la desconfianza generada a causa del fraude de clics en los sistemas de anuncios como Google AdSense. En concreto, se expone la hipótesis de que existen robots capaces de inflar el número de clics que reciben los anuncios publicados en una página web, permitiendo el lucro de la plataforma anunciadora.

Para responder a estas preguntas, se ha desarrollado una prueba de seguridad que combina técnicas de Cross Site Scripting (XSS) y de Web Crawler para superar las barreras del sistema hasta la obtención de los enlaces validados y legítimos generados por Google Adsense durante la visita de un usuario en una página web. El resultado obtenido fue positivo y en octubre de 2013 se inició una ronda de contactos con Google para advertir del agujero de seguridad detectado y colaborar en la resolución del mismo. En enero de 2014 Google no quiso afirmar ni desmentir el problema, por lo que a día de hoy sigue sin ser resuelto. El presente artículo pretende llamar la atención al respecto de este tipo de vulnerabilidades para que puedan ser resueltas y tenidas en consideración por su peligrosidad.

Metodología

La vulnerabilidad de Google Adsense se puede secuenciar en dos fases. 1) Obtener los enlaces de los anuncios originales de un sitio web de forma automática. 2) Ejecutar clics automáticos sobre los anuncios obtenidos.

1) Obtener los enlaces de los anuncios originales de un sitio web de forma automática

El objetivo de la vulnerabilidad son todas aquellas páginas web que contengan anuncios de texto de Google Adsense. En todo caso, los enlaces de los anuncios no son visibles si se observa el código fuente de la página web. Sin embargo, si se analiza el elemento web correspondiente al anuncio mediante un *navegador DOM*, sí es posible comprobar la dirección URL del enlace. Por tanto, existen mecanismos de seguridad en Google AdSense que permiten construir los elementos HTML de los anuncios mediante DOM y Java, una vez ha sido cargado el sitio web en el navegador del cliente. De esta forma, se explica que los enlaces de los anuncios no sean visibles en el código fuente de la página, aunque en realidad estén presentes. Esta medida de seguridad está diseñada para evitar que puedan obtenerse los enlaces de forma automática, considerándose una técnica inatacable según (GANDHI, M.; JAKOBSSON, M.; RATKIEWICZ, J. 2006, p.134). Por desgracia, sí es posible superar esta barrera; basta con obtener el código fuente de la página web objetivo y detectar el código de Google AdSense, véase figura 1.

```
<div id="GoogleAdSense">
<script language="JavaScript" type="text/javascript">
        google_ad_client = "pub-1229649499684927";
        google_ad_channel = "ANUNCIOS.COM"
        google_ad_type = "text";
        google_max_num_ads = 3;
        google_language = "es";
        google_safe  = "high";
        google_encoding = "utf8";
        google_ad_width = 336;
        google_ad_height = 280;
        google_ad_format = "336x280_as";
        google_color_border = "EEEEEE";
        google_color_bg = "EEEEEE";
        google_color_link = "000066";
        google_color_text = "000000";
        google_color_url = "CC0000";
</script>
<div id="GoogleAd">
<span>Enlaces patrocinados</span>
<script language="JavaScript" src="http://pagead2.googlesyndication.com/pagead/show_ads.js"
type="text/javascript"></script>
</div>
</div>
```

Figura 1. Código java de Google AdSense

El archivo con extensión de java "show_ads.js" es el encargado de embeber los anuncios en el código HTML de la página web objetivo una vez ésta ha sido cargada por completo en el navegador. No obstante, no los añade directamente, sino que introduce dos ventanas marco <iframe> a las que denominaremos "Iframe 1" e "Iframe 2", véase figura 2.

```
<!-- Iframe 1 -->
<iframe width="0" height="0" frameborder="0" marginwidth="0" marginheight="0" vspace="0"
hspace="0" allowtransparency="true" scrolling="no" allowfullscreen="true" style="display:none"
id="google_esf" name="google_esf"
src="http://googleads.g.doubleclick.net/pagead/html/r20140904/r20140417/zrt_lookup.html"></ifr
ame>

<!-- Iframe 2 -->
<iframe id="google_ads_frame1" name="google_ads_frame1" width="336" height="280"
frameborder="0" src="http://googleads.g.doubleclick.net/pagead/ads?client=ca-pub-
1229649499684927&amp;format=336x280_as&amp;output=html&amp;h=280&amp;adk=3476478377&amp;w=336&
amp;lmt=1410413197&amp;num_ads=3&amp;channel=ANUNCIOS.COM&amp;ad_type=text&amp;color_bg=EEEEEE
&amp;color_border=EEEEEE&amp;color_link=000066&amp;color_text=000000&amp;color_url=CC0000&amp;
oe=utf8&amp;flash=14.0.0&amp;hl=es&amp;url=http%3A%2F%2Flocalhost%2Fvigilante%2Fexploits%2Fexp
loit2.php&amp;adsafe=high&amp;dt=1410420397140&amp;bpp=10&amp;bdt=23&amp;shv=r20140904&amp;cbv
=r20140417&amp;saldr=sa&amp;correlator=8467094044672&amp;frm=20&amp;ga_vid=482323067.141042039
7&amp;ga_sid=1410420397&amp;ga_hid=2126412725&amp;ga_fc=0&amp;u_tz=120&amp;u_his=2&amp;u_java=
1&amp;u_h=900&amp;u_w=1600&amp;u_ah=856&amp;u_aw=1600&amp;u_cd=24&amp;u_nplug=14&amp;u_nmime=1
00&amp;dff=times%20new%20roman&amp;dfs=16&amp;adx=8&amp;ady=8&amp;biw=1600&amp;bih=795&amp;eid
=317150304%2C317150313&amp;oid=3&amp;rx=0&amp;eae=0&amp;fc=8&amp;brdim=0%2C0%2C0%2C0%2C1600%2C
0%2C1600%2C856%2C1600%2C795&amp;vis=1&amp;abl=CS&amp;ppjl=u&amp;fu=0&amp;ifi=1&amp;xpc=EvP9Cmk
S4y&amp;p=http%3A//localhost&amp;dtd=50" marginwidth="0" marginheight="0" vspace="0"
hspace="0" allowtransparency="true" scrolling="no" allowfullscreen="true"></iframe>
```

Figura 2. Inserción de iframes de show_ads.js

El Iframe 1 contiene un método de verificación de la concordancia e integridad del código de Google AdSense dentro de la página web en la que se está cargando (Linden, J.; Teeter, T. 2006). Este sistema asigna el identificador de anuncio y usuario necesarios para ejecutar el código inserto en el Iframe 2, dificultando así manipulaciones en el código fuente de los anuncios para su extracción automática. Por otra parte, el Iframe 2 carga una página web dinámica que contiene los anuncios propiamente dichos. Por tanto, para que la carga de los anuncios en el Iframe 2 sea validada y permitida por el Iframe 1, se necesita ejecutar todo el código de Google AdSense y posteriormente extraer el enlace de la página web dinámica del Iframe 2. Para ello es posible utilizar técnicas de XSS y JavaScript. En concreto, se añade a la capa "GoogleAdSense"

un formulario denominado “technical1” con un campo que almacene la dirección URL del Iframe 2. Esto se consigue mediante las instrucciones que se muestran en la figura 3.

Método de reemplazo del código fuente de la página web objetivo. Se puede apreciar cómo la capa GoogleAdSense es reescrita junto con el formulario “technical1” que servirá de recipiente de datos.

```
$html1 = preg_replace("/<div id=\"GoogleAdSense\">/", "<div id=\"GoogleAdSense\"> + Formulario technical1", $html1);
```

Formulario “technical1”, introducido dentro de la capa “GoogleAdSense” que contiene el área de texto “code1”, en el que se almacenará la dirección URL del “Iframe2”, una vez éste se ejecute en la página web.

```
<div id="GoogleAdSense">
      <div>
        <form name='technical1' action='$_SERVER[PHP_SELF]' method='post'>
        <textarea id='code1' name='code1'></textarea>
        </form>
      </div>
</div>
```

Código JavaScript para extraer la dirección URL del “Iframe2”, copiarla y pegarla en el área de texto “code1” del formulario “technical1”.

```
<script>
window.onload = init;
function init() {
document.getElementById('code1').value=document.getElementById('google_ads_frame1').src;
document.technical1.submit();
}
</script>
```

Figura 3. Código para extraer el enlace del Iframe 2

El script de la figura 3 se ejecuta una vez se carga la página web, igual que el código AdSense, pero posteriormente en el tiempo, para permitir que se carguen los Iframe 1 y 2. Seguidamente, se asigna un valor al campo del formulario “code1” correspondiente a la dirección URL del Iframe 2, identificado como “google_ads_frame1”. Una vez obtenido, se transmite el formulario “technical1” con dicha información.

En la figura 2 se observa que la dirección URL del Iframe 2 no es correcta, puesto que introduce entre sus variables la dirección del dominio desde el que se está ejecutando el código: obsérvese la URL “http://localhost/vigilante/exploits/exploit2.php”. Si se hiciera clic en un anuncio sin modificar estos valores de páginas referentes, Google AdSense podría detectar algún tipo de incoherencia. Para evitarlo, la dirección URL del Iframe 2 debe ser modificada en las variables **&url** y **&p**, véase figura 4.

```
$code = $_POST[code1];
$code = preg_replace("/\&url=.*\&adsafe/", "&url=http://www.anuncios.com/&adsafe", $code);
$code = preg_replace("/\&p=http.*/", "&p=http://www.anuncios.com", $code);
```

Figura 4. Modificación del enlace del Iframe 2

Cuando el enlace del Iframe 2 es recuperado, se reemplazan las variables **&url** y **&p** por la dirección del dominio de la página web que se está estudiando, en este caso *http://www.anuncios.com*, mediante expresiones regulares. Una vez la dirección URL del Iframe 2 está preparada, se recupera su código fuente, mediante la función “file_get_contents”,

accediendo directamente a los enlaces de los anuncios ya validados por Google AdSense. Durante este proceso, se puede emplear DOM para cargar la estructura de HTML de la página y mediante XPath para obtener únicamente los enlaces disponibles, véase figura 5.

```
$html2 = file_get_contents("$code");
$dom2 = new DOMDocument();
@$dom2->loadHTML($html2);
$xpath2 = new DOMXPath($dom2);
$links = $xpath2->query("/html/body//a");
```

Figura 5. Obtención del código fuente de la página del Iframe 2 y recuperación de los enlaces de los anuncios

El resultado de aplicar este método son los enlaces de los anuncios, tal y como se presentan en la figura 6. Para comprobar su funcionamiento, este programa puede ser descargado en: http://www.mblazquez.es/docs/google-ads-extractor.zip. También puede visionarse su funcionamiento en: https://youtu.be/0tIBcJ-VN7s

```
http://googleads.g.doubleclick.net/aclk?sa=l&ai=CCWOcql4RVJDHLIHCiQa1zoCICa-
GqZsH75Pj9aIBwI23ARABIPmY_AEoA1DA6qP9-
v____8BYNW11wKgAdHQoN4DyAEBqQKMWfG0Fae1PqgDAcgDwwSqBG5P0Ig3k_RIM7eChCXVjUsXttfDLlj6od6JU7FeWhd
mg0GwM8-w6Nhuh9awpyuPhzyt-gKK2kLj6_Fp03lAl8FoHzX6xTk8nztGi79o9q9viXDkav02-Yi2F5OT67hR-
ItZO2J04gvpcTvvajkQpYAHl6_fIQ&num=1&sig=AOD64_2umGsoLzRoV9h4YazaUGzHORn0xQ&client=ca-pub-
1229649499684927&adurl=http://candidatos.sanroman.com/resultado-
busqueda.php%3Ffiltro%3D%26area%3DMarketing%26sector%3D0%26modalidad%3D0%26provincia%3DMADRID%
26seccion%3D

http://googleads.g.doubleclick.net/aclk?sa=l&ai=Cj0h3ql4RVJDHLIHCiQa1zoCICfGSgIoF2Z2bkJcBwI23A
RACIPmY_AEoA1DwkovEBGDVtdcCoAH___T8A8gBAagDAcgDwwSqBG5P0NgMkPRLM7eChCXVjUsXttfDLlj6od6JU7FeWhd
mg0GwM8-w6Nhuh9awpyuPhzyt-gKK2kLj6_Fp03lAl8FoHzX6xTk8nztG1dsY_a9viXDkav02-Yi2F5OT67hR-
ItZO2J04gvpcTvvEC0ehYAH6f-KAw&num=2&sig=AOD64_2PS3pUS7ccIZ8gg9nxsoRo1dgCCg&client=ca-pub-
1229649499684927&adurl=http://es.emailbrain.com/ebs/index.shtml%3FMedium%3DPPC%26Campaign%3DES
_Email_Marketing%26Adgroup%3DEmail_Marketing_-_Broad

http://googleads.g.doubleclick.net/aclk?sa=L&ai=C4_Hlql4RVJDHLIHCiQa1zoCICamd6N0E4def75EBwI23A
RADIPmY_AEoA1CJtf3UB2DVtdcCyAEBqQKMWfG0Fae1PqgDAcgDwwSqBGtP0Khal_RKM7eChCXVjUsXttfDLlj6od6JU7F
eWhdmg0GwM8-w6Nhuh9awpyuPhzyt-gKK2kLj6_Fp03lAl8FoHzX6xXE8Vwfifvy9qq9vw3DkOuA2-
n2112aQK0xS6H9aK9d34qvv2A9jrYAH2Zf5Lg&num=3&sig=AOD64_1aYwxLbhcZISbWwRtDQhgLRfddmg&client=ca-
pub-1229649499684927&adurl=http://www.banderasysoportes.com
```

Figura 6. Enlaces de anuncios obtenidos automáticamente

El método de extracción automática de enlaces de anuncios AdSense puede resumirse en los siguientes pasos: a) extracción del código fuente de la página web objetivo, b) detección del código de Google AdSense, c) Ejecución del código de Google AdSense, d) Carga de Iframe 1 y 2, e) Inserción del formulario technical1 y campo code1, f) Extracción de la dirección URL del Iframe2 al campo code1, g) Envío del formulario al mismo entorno de ejecución, h) Preparación de la URL del Iframe2 transmitida en el formulario, i) Obtención de código fuente de la dirección URL del iframe 2, j) Extracción de enlaces de los anuncios.

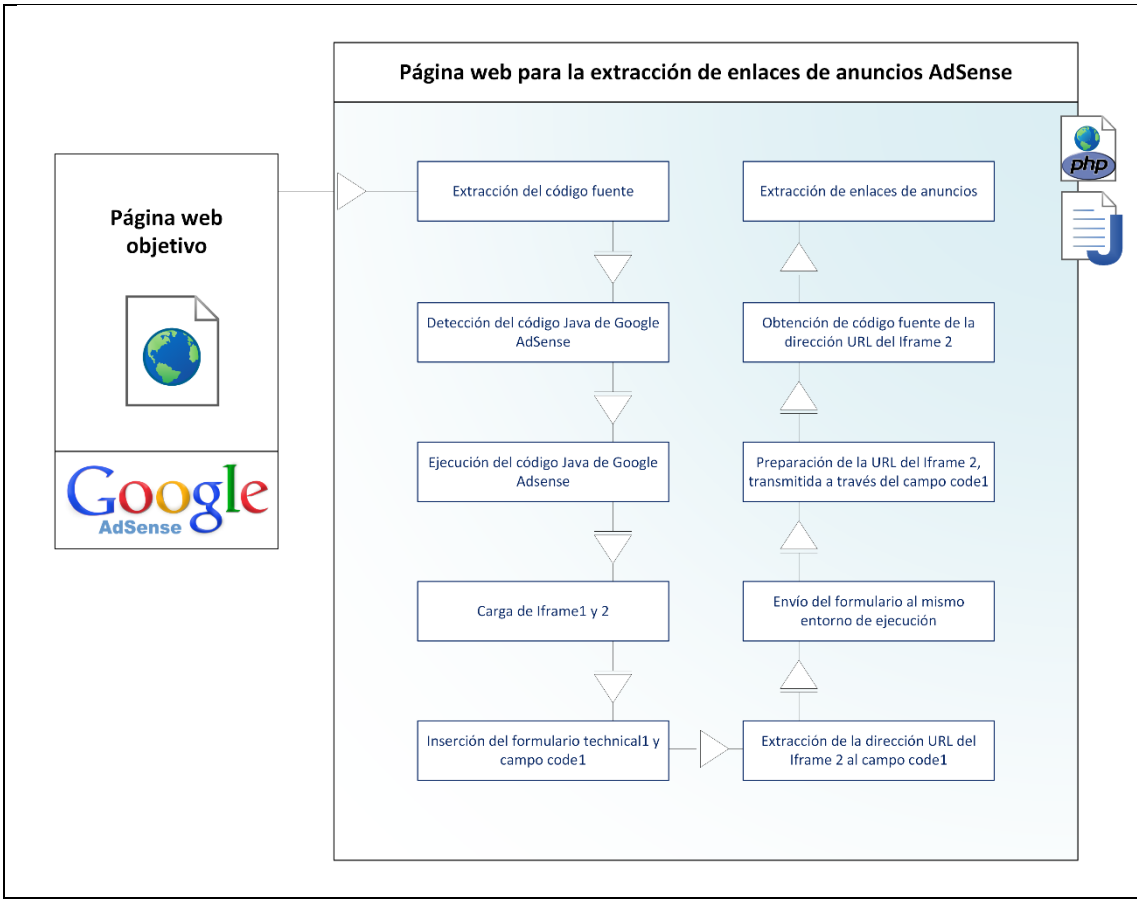

Figura 7. Resumen del proceso de extracción automática de enlaces de anuncios AdSense

2) Ejecutar clics automáticos sobre los anuncios obtenidos

Demostrada la capacidad para extraer los enlaces de los anuncios de forma automática, no resultaría raro realizar clics en los enlaces de los anuncios que se obtengan. Esta tarea puede resultar compleja, ya que Google AdSense viene introduciendo medidas de seguridad estadística que pueden llegar a reconocer clics fraudulentos. Éste sistema de filtrado (Google Support. 2014) es capaz, hipotéticamente, de reconocer aquellos clics destinados a incrementar los gastos de publicidad para aumentar los beneficios de los propietarios de sitios web que alojan sus anuncios, provocando en consecuencia su inhabilitación automática. Según (Adwords Blogspot. 2007) cerca del 10% de los clics registrados son fraudulentos, reconocidos automáticamente por el sistema, quedando un 0,02% de casos que requiere una investigación especializada para determinar su validez. También cabe señalar la posibilidad de los falsos positivos en el proceso, tal como indica (Dave, V.; Guha, S.; Zhang, Y. 2012, p.178) en su trabajo sobre la cuantificación del click-spam en las redes de anuncios.

Profundizando en la detección del fraude de clics, según (Kshetri, N. 2010), los sistemas actuales de seguridad pueden clasificarse en: a) Detección de anomalías, b) Detección heurística de clics no válidos en función de determinadas variables como la proporción, ratio de clics, trama de navegación. c) Mediante clasificadores de patrones de comportamiento de usuarios y clics. En el caso de Google AdSense, es muy probable que incorpore una combinación de tales métodos en base a las patentes publicadas al respecto.

Por ejemplo, el método de identificación del procesamiento de anuncios (Li, Z.; Ou, C.; Park, S.U.; Savoor, R.; Sposato, S. 2007) diseñado para testar la IP y el tiempo de permanencia del usuario en la página web en la que realiza el clic. Si el clic es automático, el tiempo de permanencia del usuario sería prácticamente nulo, o bien la IP que realiza clics es siempre la misma.

También figura la patente del método de detección de clics fraudulentos de (Gillespie, J.; Meggs, A.F. 2007) que registra cada clic del usuario y su trazado de búsqueda y navegación, añadiendo información como la tasa de clics por minuto, el ratio de cobertura de clics, promedio de visitas y clics del sitio web, ratio de visitas y clics de los usuarios, entre otros. Este método supone que una desviación superior a la normal, con respecto a las estadísticas de crecimiento de visitas y clics de un sitio web, implica la detección automática de clics ficticios. La debilidad de este método, radica en la posibilidad de camuflar estadísticamente los clics, incrementando el número de visitas y controlando a la par los tiempos de permanencia en las páginas web, antes de ejecutar un clic automático.

La patente de detección de fraude de clics de (Zwicky, R.K. 2010) plantea un sistema por etapas que tienen en consideración al anunciante en la detección de fraudes, dotándolo de paneles de supervisión, con los que puede denunciar situaciones sospechosas, como por ejemplo direcciones IP de visitantes considerados como robots, incrementos de clics repentinos o aberraciones en la navegación del usuario que hace clics.

Sobre los métodos de detección de clics fraudulentos mediante clasificación automática, es clave la patente de (Yan, J. H.; Jiang, W. R. 2014). Presentan un sistema que registra mediante etiquetas a los usuarios que acceden a la página web objetivo, identificando su navegación y clics sobre anuncios. Según este método, si se obtienen patrones que no coincidan con el uso habitual del sitio web, o bien en relación a su número de visitas, el usuario pasa a ser considerado robot y sus clics como fraudulentos. Este hecho queda registrado y sirve de experiencia para la detección de nuevos clics malintencionados.

Considerando todas las medidas de seguridad descritas, su completa vulneración puede resultar un proceso difícil de determinar sin la colaboración de Google. A pesar de todo, podrían hallarse problemas de seguridad, si se cumplen las siguientes premisas:

a) **Grupo de páginas web para la captación de usuarios.** Una manera de evitar que las direcciones IP de los usuarios sean siempre las mismas es utilizando un grupo de páginas web con afluencia de visitas regular y variada de todo el mundo. Cada visita puede utilizarse para cargar en una ventana marco iframe el programa necesario para extraer los enlaces de los anuncios de una página web objetivo y hacer clic en ellos. Por tanto, el factor tiempo y dirección IP del usuario serían resolubles. Por otra parte este método no es infalible, ya que Google puede detectar que el número de visitas del sitio web objetivo que posee los anuncios no es el habitual y sin embargo registrar más clics de los inicialmente probables. En tal caso, es posible redirigir visitas sin que éstas efectúen clics sobre los anuncios, para que de forma variable y progresiva, aumenten el número de visitas del sitio y con ello el tráfico. Si el tráfico aumenta, también la posibilidad de que proporcionalmente se realicen clics sobre los anuncios en el sitio web.

b) **Programa de re-direccionamiento, extracción de los enlaces y clic automático.** Un programa JavaScript puede generar una ventana marco iframe invisible sobre las páginas web destinadas a la captación de usuarios, que permitiera cargar a su vez la

página web objetivo que contiene los anuncios en los que se pretende hacer clic y a su vez ejecutaría el método de extracción de enlaces propuesto anteriormente, para finalmente decidir en virtud de las estadísticas sí se debe hacer clic sobre el enlace del anuncio o bien se debe esperar o bien no hacer dicho clic. Este sistema podría vulnerar los métodos heurísticos, patrones y clasificación automática, al adquirir hábitos propios de los usuarios normales del sitio web.

Estos problemas de seguridad se traducen en una estrategia de vulneración para realizar clics automáticos, véase figura 8.

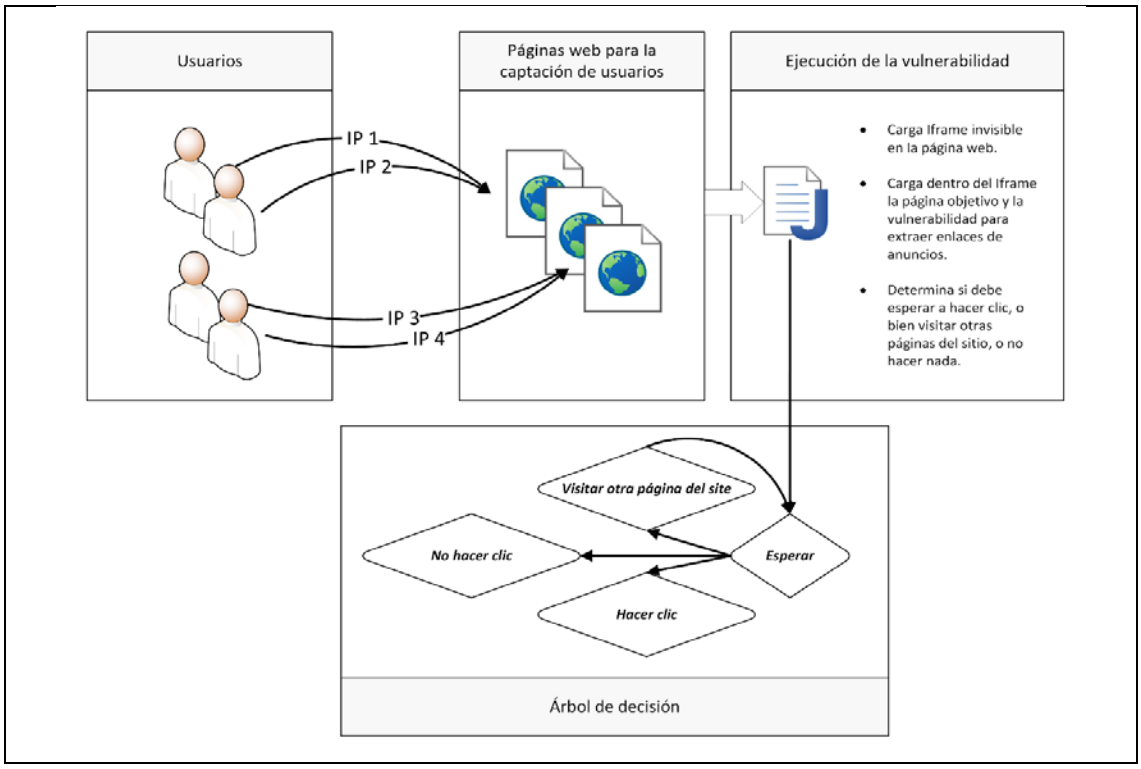

Figura 8. Estrategia de clic automático

Los usuarios que acceden con diversas direcciones IP, visitan páginas web que "secuestran" su sesión de usuario a través de la ejecución de un código Java que carga un iframe invisible, en la página que están visitando. Sin embargo, en el iframe invisible se carga la página web objetivo con los anuncios y se ejecuta a su vez el programa de extracción automática de los enlaces de sus anuncios. Una vez almacenados, el programa puede ser diseñado para tomar diferentes decisiones, por ejemplo; a) Esperar un tiempo aleatorio o definido estadísticamente, b) Visitar otra página del sitio web para simular la consulta del usuario, c) No hacer clic según el ratio del sitio web, d) Hacer clic sobre un anuncio. La vulnerabilidad se ejecutaría tantas veces como usuarios accedieran a las páginas web de captación, obteniendo para cada uno de ellos anuncios diferentes, personalizados para cada perfil, de la misma manera que lo haría Google AdSense sin vulnerabilidad.

Para probar la hipótesis de la estrategia del clic automático, se ha desarrollado un programa que simula el supuesto de un blog de captación de usuarios, denominado "blog1" y un blog objetivo que contiene anuncios denominado "blog2". El blog1 ejecuta el código "google_analytics_top.js" que permite introducir en la capa con identificador "*booster*" otras dos capas denominadas "sub1" y "sub2" así como un nuevo iframe que contiene el código para la extracción de los enlaces de los anuncios en el blog2 y el proceso de carga de los mismos, en

virtud de las reglas programadas y patrones que simulen un comportamiento normal del usuario. Una vez hecho el clic sobre el anuncio correspondiente, el programa elimina todo rastro de presencia eliminando la referencia al enlace que lo contenía en la capa. Este programa puede ser descargado en: http://www.mblazquez.es/docs/booster.zip .

Conclusiones

1. Google AdSense ofusca el código de los anuncios, dificultando el acceso a sus enlaces, para evitar que éstos sean utilizados con propósitos maliciosos. Para ello evita visualizar el código fuente de los mismos por medio de dos iframe embebidos y generados a partir de "show_ads.js". De esta forma, los robots especializados no pueden recuperar automáticamente los enlaces de los anuncios.

2. Sin embargo, la seguridad de los anuncios publicados en los sitios web afiliados al programa de Google Adsense está potencialmente comprometida, ya que es posible recuperar automáticamente sus enlaces junto con el código de validación de Google, tal como se ha demostrado. Esta técnica podría ser aprovechada para aumentar los beneficios en el sistema de pago por clic o bien conseguir la inhabilitación de las cuentas de las plataformas de publicación de anuncios de la competencia, si se superan sus límites estadísticos.

3. Google AdSense no protege tan bien los enlaces de los anuncios como los sistemas de detección a posteriori. Esto significa que los clics ficticios y verdaderos tienen que ser filtrados a posteriori. De hecho, Google emplea hasta 5 patentes especializadas en la detección estadística de clics fraudulentos.

4. Teniendo en cuenta que los sistemas de detección por clic fraudulento se basan en los patrones normales de navegación e interacción del usuario, éstos podrían ser superados si se aplicara un método de captación de usuarios y re-direccionamiento, combinado con la extracción automática de anuncios. Es posible simular la navegación del usuario en una página web y efectuar clics de forma simple o bien de forma coordinada de acuerdo a las estadísticas que se deseen.

Bibliografía

Google Support. 2014. Invalid clicks. In: Google Adwords. Available in: https://support.google.com/adwords/answer/42995?hl=en

Adwords Blogspot. 2007. Invalid Clicks – Google´s Overall Numbers. Available in: http://adwords.blogspot.com.es/2007/02/invalid-clicks-googles-overall-numbers.html

Yan, J. H., & Jiang, W. R. (2014). Research on Information Technology with Detecting the Fraudulent Clicks Using Classification Method. *Advanced Materials Research*, *859*, 586-590. Available in: http://www.scientific.net/AMR.859.586

Gillespie, J., & Meggs, A. F. (2007). Click-fraud detection method. *U.S. Patent Application 11/648,576*. Available in: http://www.google.com/patents/US20080162475

Li, Z., Ou, C., Park, S. U., Savoor, R., & Sposato, S. (2007). System and method of processing online advertisement selections. *U.S. Patent Application 11/800,966*. Available in: http://www.google.com/patents/US20080281941

Linden, J., Teeter, T. (2006). Method for performing real-time click fraud detection, prevention and reporting for online advertising. *U.S. Patent Application 11/258,977*. Available in: http://www.google.de/patents/US20060136294

Mann, C. C. (2006). How click fraud could swallow the Internet. *Wired Magazine*. Available in: http://archive.wired.com/wired/archive/14.01/fraud_pr.html

Kshetri, N. (2010). The Economics of Click Fraud. *IEEE Security & Privacy*,*8*(3), 45-53. Available in: http://libres.uncg.edu/ir/uncg/f/N_Kshetri_Economics_2010.pdf

Gandhi, M., Jakobsson, M., & Ratkiewicz, J. (2006). Badvertisements: Stealthy click-fraud with unwitting accessories. *Journal of Digital Forensic Practice*, *1*(2), 131-142. Available in: http://www.iu.edu/~phishing/papers/gandhim.pdf

Zwicky, R. K. (2010). Click fraud detection. *U.S. Patent No. 7,657,626*. Washington, DC: U.S. Patent and Trademark Office. Available in: http://www.google.com/patents/US7657626

Dave, V., Guha, S., & Zhang, Y. (2012, August). Measuring and fingerprinting click-spam in ad networks. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication* (pp. 175-186). ACM. Available in: http://dl.acm.org/citation.cfm?id=2342394